Design for Internet of Things

Prof. T. V Prabhakar
Department of Electronic Systems Engineering
Indian Institute of Science, Bengaluru

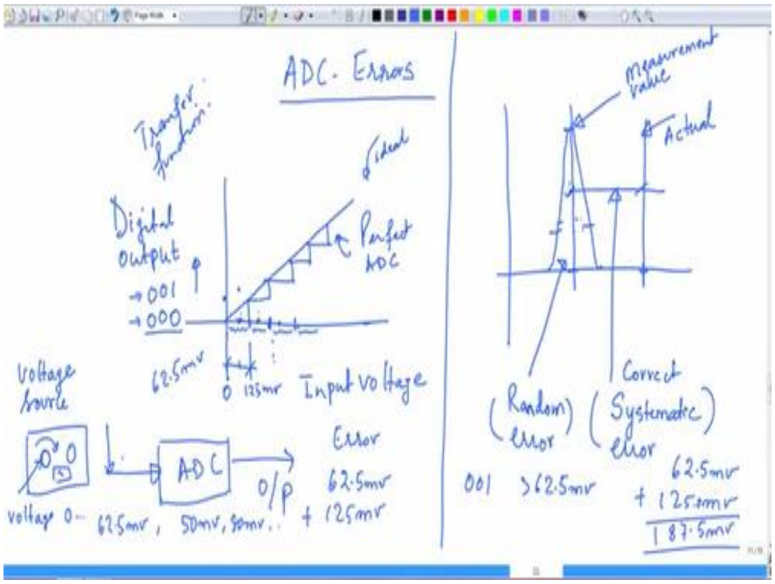
Lecture - 06
Challenges Part-04

Welcome back. We will look at ADC errors and we will try to get a comprehensive view of this aspect of ADC because in IoT we will be sensing. So, the sensor is connected to an ADC port and if you do not understand the functioning of the ADC and errors associated with ADC your values that you get for any sensing application and sensing, and control application can be disastrous.

So, particularly industrial IoT or medical IoT sensing value should be perfect, if you look building a ventilator and you are looking at the exact pressure value of which are sensing it should be prefect. So, that you cannot blame you cannot say that I bought the best sensor, but I did not get a good performance from the electronic system. It is because of your limitation to understand how the ADC works, how it actually converts.

Now as IoT engineers, what are the things? What are the sources of this error? You are to understand. This is a very clear. In fact, this error and error budgeting and working with errors is a reality. There is nothing like ideal anywhere. If you take a picture like this.

(Refer Slide Time: 01:59)



Let us see I just give you a I mean this is something that I was seeing somewhere and then I thought this will be nice to introduce even here. Let us say this is the true value of something, this is the true value, this is the actual value of the sensed output, the actual value. But when you actually did all, the sensor you know, you took the sensor connected to the ADC and you connected, and you looked at the output your value was here.

So, this is measured. So, actual is this measured means when I say I have also included the error in the measurement already. Measuring equipment also can have an error plus you connected to ADC, it will also give you a certain error. So, we will factor all of them slowly. This is your measurement value measured, by a measurement you got this value. So, this is actual measurement value.

So, first time when you measured you got this value and you got this value, but then when you did another measurement, this line perhaps shifted to this point. Another time when you did a measurement, this line perhaps shifted here. Another time when you did a measurement it perhaps went here and when you did another measurement, it went here. So, you can go on doing these measurements.

And you can arrive it so, let me put it back. You did all these measurements repeatedly. And what you got is not a single line here, but you got a distribution of errors. In other words, you could see that this is now become like a mean and there is a distribution of errors here. So, and this is usually Gaussian and so on, so this is what you got. Now you can say this is the error which I can easily correct.

I can correct this, because if I know this line, I know my actual line, I know this line, I can more or less, corrected for it. So, that I can add some value, some offset, some bias, something I can do, and I can correct this measurement value that I got here. This kind of fixed line or fixed distance between the actual and the measurement which you can correct because you are able to measure and you can quantify it and corrected clearly indicates this is called Systematic Error, systematic errors can be corrected.

What you have difficulty is this part here, all these regions you do not know where exactly that value comes, because this is random. Therefore, this is Random Error. So, any ADC will have systematic error and will have random error and you can only you know get look at data sheets and understand all the error values which are mentioned in the ADC understand them carefully, you may not want to, you know sort of worry about it, because any ADC, there is no ideal ADC.

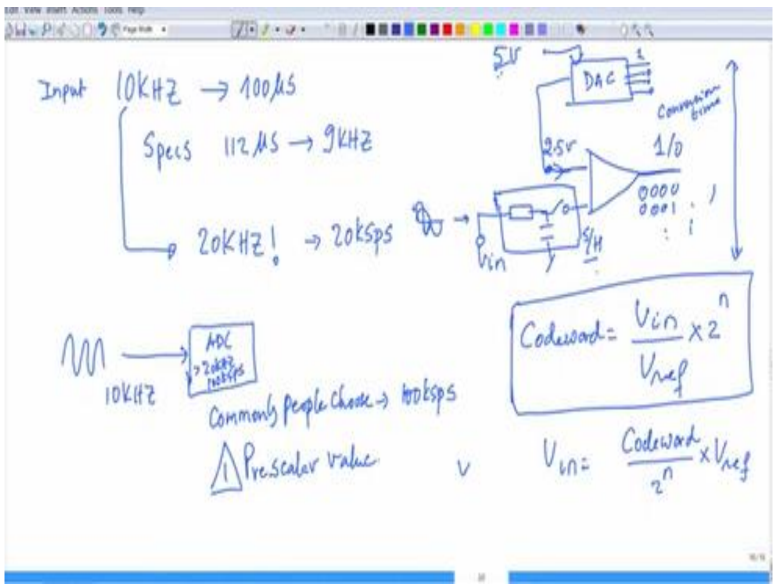
There is no ideal ADC at all bound to have a lot of errors, you should know how to work with them those errors. So, this is a sort of a I would say this picture on the right side is more of a general understanding, we will try and see how to map it into the ADC. But in general, any error you should first be clear that you can do those errors which are correctable, which you can measure, and you know sort of apply a certain offset.

And you can correct them, which are essentially systematic errors and then there are random errors, random errors are difficult to catch. Now let us come back to this picture on the left side. So, I will draw a line here and we will just worry about this line. See, if you apply a certain input voltage to an ADC, this is your ADC, this is what you are applying here is analog input, what you are getting is a digital output.

What I did here was I plotted the input versus the digital output. This is also called often they call it by a name called Transfer Function. You can look up there a lot of material related to transfer function, but I will try to give you a very intuitive understanding because you need that as a basis for all your design system for your design and choice of components and so on. You need to look at those data sheet parameters.

If you take an ideal ADC any small input that you apply here you will have a corresponding digital output, any small variation any small input you will have. So, in a way this line will become continuous. You apply a voltage here you will get one point; you apply a very small voltage here next to that infinitely small increment over the existing voltage you will get one digital output. That kind of ADC is very ideal it does not exist at all. And it does not exist because of what we discussed previously. Look at this picture.

(Refer Slide Time: 08:40)



So, what actually happens is when the input is coming here, right here, AC input. So, let me remove this and show it like this much higher frequency comes and so on. So, this frequency input, high input signal is coming here this capacitor has to charge and I mentioned to you that the output has to hold when the capacitor is red, so that you get the digital output. Now this has a certain latency associated with the operation. All this actually means that there is something called a conversion time.

There is a conversion time and then there is also so closely associated the conversion time is the sampling rate. So, you can sample it only at a finite rate and that depends on the sample and hold circuit, the impedance offered by the ADC and so on. So, many parameters are associated with it. Anyway, abstracting all that simply means if there is a finite time over which the ADC takes to convert for the input to the output and there is the sampling rate.

The minimum and maximum sampling rate that the ADC supports very high frequency signals, input voltages will keep changing. So, at very high rate, that signal is now lost that the next level of voltage is actually lost. There is no way by which you can you know convert, that small voltages which are arriving. Therefore, what it boils down to is that you cannot have an ideal ADC line like this it is not going to work at all.

So, what you will have is some input voltage up to one point any voltage up to this point will have a corresponding digital output, this will be the code word output. So, then what happens? Up to the next point it will remain the same code word. Any voltage here is still represented here only it is still represented by this point only. And once it touches this point the input voltage comes here you will get a corresponding code word output and then it goes on like this.

So, the next point is here and so on. So, up to from this point at this point it is all represented by this value only and from here onwards whatever is increased in voltage will still be represented like this and then like this. So, think about a staircase, you are climbing a staircase and you are moving all the time on the staircase. So, the point is this is ideal, and this is also a perfect ADC in a way.

This is not ideal, but perfect it is doing what it is supposed to do. The point is you do not have perfect ADC is also, that is a problem that is why I have shown this right-side picture. If you had a perfect ADC, you would have no issues at all; everything is same in a way. Also let us consider an example of to understand the transfer function by taking a reference voltage of one volt. If you take one volt as the Vref, the first code word generates for Vref of one volt into 2 power n and you want to know the first code word.

So, this will be 0.125 it will be 0.125. If you have 0.125 volts you will get the first code word, you can check this. The second code word will come when it is anywhere between 125, 0.125

into two divided by one volt into two power n which is again cube in our case you will get the second code word. So, now if you now map this into this picture, let us put it here. Any voltage from 0 volts to 125 milli volt will simply read all zeros, the very first code word which is 0.

Then the second one is from 125 to 125 into 2 which is 250 millivolts this to up to250, from 125 to 250 millivolt it will read the second code word. And the second code word corresponds to digital output of 001. So, this is how it should work. Take a voltage source, power supply for instance it is a voltage source. Simply power supply put it to zero knob, put into complete shift the knob to complete zero volts there are two knobs you can do current limit adjustment and voltage adjustments.

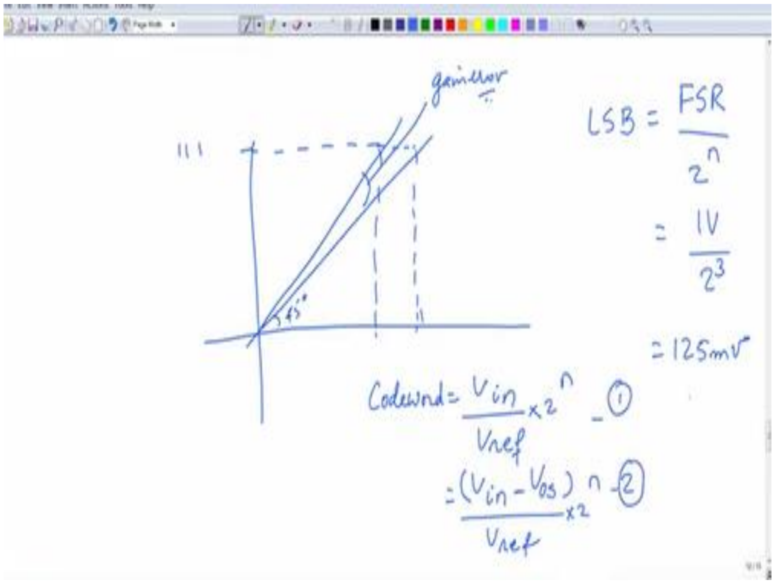
So, this is your voltage knob. If you change this knob to the right, the voltage increases there is a small meter which will tell you, there is a meter here. This meter with graduation will show you increase in the voltage slowly it will keep changing. Now supposing you start from 0, and you start slowly moving this knob to the right, you will find to your surprise that the first digital output or the 0 volts you leave that aside, the first code word output which is 001.

That actually occurs not at 125 millivolts, but it occurs much early. In fact, if you move the knob to 62.5 millivolts or there about, I am being very careful 62.5 millivolt or there about. You may already see 001 that is the point. I use the word there about because I am not sure you are also not sure for that ADC, whether it will trigger at half of 125 millivolt which is 62.5 or whether it will trigger to the second code word at 50 millivolt or whether it will trigger at 80 millivolts, we do not know this.

We do not know this for sure, but for sure, it will never trigger at 125 millivolts, it might trigger much earlier. It is also possible it may trigger even later, why should it trigger at 125 millivolt, it may also trigger at 130 millivolt, may trigger at 140 millivolt you really do not know till you actually make a measurement. So, you need a multimeter you need a digital output, and you should be able to read those values which essentially means it is not an ideal perfect ADC at all.

Now, how do you account for all this? But one thing you can tell for sure, if it triggers at let us say at 62.5 millivolt the next trigger will definitely be after 125 millivolts that you can be sure. To some extent you can say that that additional point will come after periodically after every 125 millivolt. Now go back to this picture to the right-side picture. This clearly is a systematic error; you have a systematic error. The systematic error can be easily corrected.

(Refer Slide Time: 17:56)

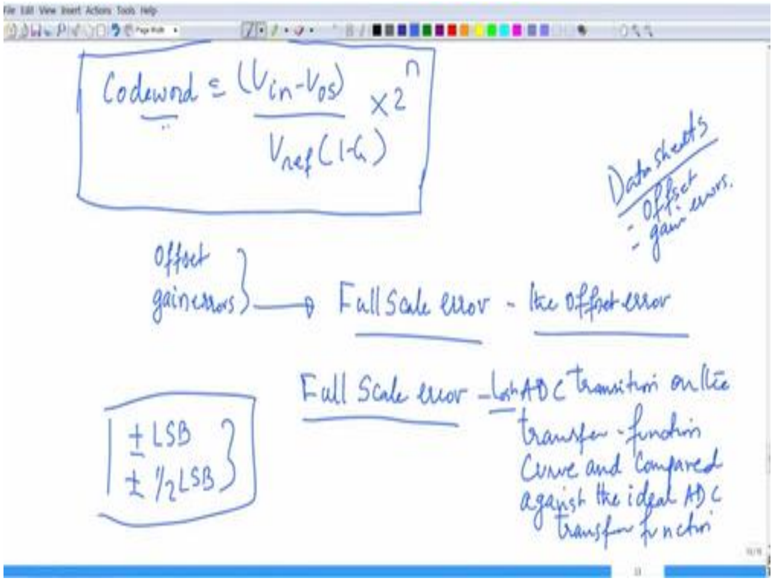


Therefore, it is easy for you to modify this expression which is code word is equal to Vin divided by Vref into 2 power n can easily be modified as Vin minus Voffset divided by Vref into 2 power n. This can be easily said you are subtracting that. This is something that you can already factor in for input offset error, one of the first systematic errors. Similarly, I will give you a feel for one more type of error that cross in.

And that is so, now we discussed about this side; you took a multimeter from zero. There is occur problem as you move on towards the maximum voltage. The maximum voltages as you are moving up the ladder, we took this as one volt as I mentioned to you. You will actually get the final code word, which is corresponding to digital value 111, much early that also is another issue for you.

In other words, this infinite resolution 45-degree curve can actually go a little more steeper. And it might reach to a voltage which is much before the actual voltage that should have triggered the output voltage. This is called gain error; this is another problem. Again, this gain error can be factored into the code word because it is also a systematic error. So, people use a convention of modifying so, this is your expression one, this is your expression two and the third expression is including the gain error.

(Refer Slide Time: 20:46)



So, let me put expression three which is code word is equal to Vin minus Voffset, Vref into 1 minus gain times 2 power n, but I will tell you that the gain terms comes here. So, this is the final expression and the code word itself that you are getting, which is nothing but corresponding to the digital value, the digital value as seen in this transfer function. This is digital value, as seen from this transfer function is clearly triggering at the lower side relates to offset error and the issue of arriving at the final code word ahead of the actual voltage is indeed the gain error.

These are the two things which are systematic by nature. So, this is the final expression which included the offset error as well as the gain error. Now, there are some formal definitions which you have to note. So, for example, the definition of gain error is typical of what people say is the error associated with the data converter, which is full scale error minus the offset error, this is the gain error.

So, gain errors are typically defined as full scale error minus the offset error. What is full scale error? How do you do that? It is measured as the last ADC transition on the transfer function curve and compared, this important against the ideal ADC transfer function so, this is important. So, gain error is defined as the full-scale error minus the offset error and the full-scale error itself how do you measure it?

Last ADC transition on the transfer function curve and compared against the ideal ADC transfer function. So, these are important points that you will have to look up in the data sheets, will talk about the offset errors and the gain errors. Now that is these two parameters are insufficient for you, you need to also look up at the other type of error and I will show you what the other errors are.

Normally if you consider this picture, we wrote here some numbers 0, 125 millivolt and so on which is essentially our example of taking FSR as one volt and three-bit resolution then we get this 125. This is from zero to 125, we said the transition is from 000 to 001 that is the next code word. Usually, they do not express it the community does not express in terms of voltages, they usually express in terms of LSB's.

So, people talk about how many LSB’s? Because if the reference changes, if your FSR voltage changes, if your Vref changes this voltage has no meaning. It will if it is 5 volts It is something else if it is one volt it is 125 as we saw. Therefore, when people talk about the steps and talking about errors, people talk in terms of LSB errors, the number of LSB is associated with.

Therefore, you can see that an ideal ADC, you can never have because you need infinite resolution. And this is a resolution which we have seen, which is like a step errors associated with ADC people refer to them as plus minus of LSB or plus minus of half LSB. It is easy to connect, you do not want to refer in terms of voltage, but you want to refer it in terms of LSB. So, that is the usual convention. So, now let us go back to the datasheet.

(Video Starts: 27:28)

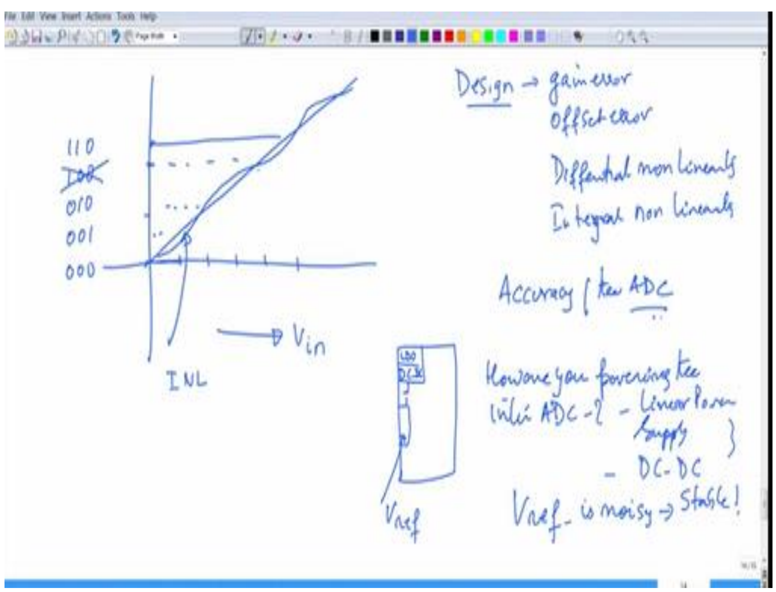
Now, it is easy as you can see that this data sheet can be understood better in the following way. The ADC of interest is a 12-bit resolution, it has plus minus one LSB of DNL, DNL stands for Differential Non-Linearity. And integral nonlinearities also plus minus one LSB then other parameters associated with that are also mentioned to you, as we go along, you will see in this data sheet that several things would be clearer as we move on. So, let us see the other data sheet parameters. You can see here the offset error.

Look at just this DC accuracy, integral non-linearity error is expressed in typical, there is a typical value and there is a maximum value of plus minus one LSB, differential non-linearity error is plus minus 0.5 a typical and maximum of plus minus one LSB. No missing codes over temperature, what is this? So, you have to be you have to be careful about understanding what this actually means. That means there are some missing codes? The answer is yes.

Actually, there are some missing codes, and we will come to those missing codes. Anyway, before we move on offset error plus minus 3 LSB maximum typical is plus minus 1.25 of an LSB. And then gain error is plus minus of 1.25 and a maximum of plus 5 LSB. Before we understand our immediate attention goes to what is this missing codes?

(Video Ends: 29:37)

(Refer Slide Time: 29:42)



So, I will show you what this missing code problem is when we come to the differential nonlinearity in the case of ADC, this is your ideal. And you would essentially have this is your transfer function which we know it so well now. This is your Vin, input voltage and these is your digital output and simply say output. And this is so I will just move this arrow so that we will be able to accommodate 001 and I will also move this now that we know it is and so on.

See, what actually happens is the ADC is performance what would happen? Ideal, let us go back to the ideal case. Till here, it will be all zeros and then it will transit to let us move this a little bit. So, that we can map it to 001 here and then it will remain then come up again here and up here is what we know very well now and so on. If an ADC is not performing up to the mark, at this point, what would happen is? It should have triggered here; this line will be missing.

What is the trigger is this instructs triggering the previous code 010, 010 then it should be 100 and 110. This code is gone it will never give you that. This is completely what actually happened was the input voltage changed in a normal way but ADC so nonlinear that it skipped one digital output. So, skipping of digital output is also possible that need not happen here. It can happen here for some ADC it can happen here some made is it trigger happened here.

So, it is all random, input would have changed but code output would not have come, it would have skipped that and gone to the other one. This problem also occurs, and this is nothing but the differential nonlinearity. Similarly integral non-linearity, integral is nothing but accumulation of the differential nonlinearity itself. Again, it is good to understand this pictorially. So, let us do that pictorially.

So, if you remove all this and look at the transfer function curve, some places it may go low, some places it may go high may go like this. This is integral nonlinearity, this is your I which you saw it here, integral non-linearity just thing but accumulation. So, now you should know how to design with gain error, offset error with known DNL and integral nonlinear. I would like to leave it at this stage because this is a basic course on ADCs which should not delve into too many details.

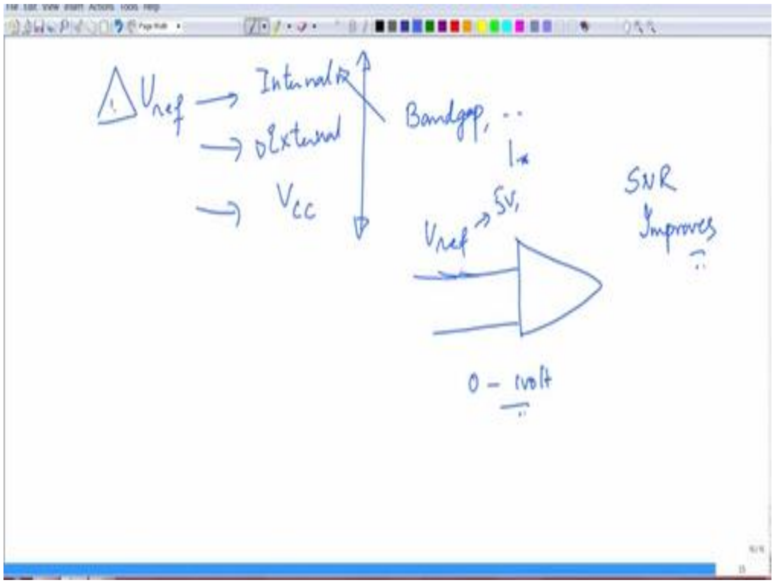
But at the same time this is important to know. There is one more important parameter which you must know which actually relate to the accuracy of the ADC. And that point I would like to discuss with respect to an important picture here which is referred to The Vref. See, this Vref also you have to put a lot of attention pay a lot of attention to Vref. Because the;Vref is the one that is being used for the comparator operation if the Vref changes, the output will change.

Now, you have compounded the problem by the code output that you are getting, the digital value you are getting is not because of any nonlinearities which you have any way you are to battle. The errors associated with the actual component, but you have to battle the voltage that you have applied to that. Why is that important? It is important because, when an ADC is powered in an IoT device.

An IoT device SoC has a power supply section which may contain a linear regulator, or it can contain a DC-DC regulator. Now, the ADC is receiving power from one of them either this or from this, one of them. Now, if it is receiving power from DC-DC, there is a lot of switching noise which is associated with DC-DC systems. But in LDO, it is a lot more smoother. Now, if Vref is applied internally to this ADC and if Vref is fluctuating randomly then the output ADC output also will change, this is one problem.

So, you have to be careful on how are the powering the internal ADC. Is it through linear power supply or is it through a switching power supply? This is one part. Second part is also about the fact that if Vref is noisy because this essentially means if Vref is noisy output will be unpredictable again. So, you need a Vref which is stable. Usually if you take Arduino and other controller boards that you know very well.

(Refer Slide Time: 37:45)



Pay a lot of attention. So, I will put one mark here pay attention to Vref, you can configure in internal you can configure external, or you can configure it from coming directly from Vcc. Any of these three things are possible although know code has specific settings for you to choose, so that you can keep the Vref as stable as possible. For any accurate measurement, it is good to use very accurate stable voltages, which you can feed into an ADC from outside that is called external Vref.

If you are choosing from within the chip, choose internal but choose something which is also very stable, which will give you a very stable reference. Inside the chip, they usually use what is known as a band gap reference. So, you try to choose the band gap reference as much as possible. The only issue with band gaps of our internal references are in fact even that is true for Arduino internal reference. I think the internal reference of Arduino is a little over one point some voltage I think 1.1 or 1.2 volts, I do not recall correctly so, it is a little over one volt.

So, in other words, any comparisons that you are doing is typical of the sensor output giving you anywhere between zero and one volt, keeping your reference at one volt. So, you have to think about what is your sensor, what is your sensing parameter and what is it that is giving from its lowest value to the highest value and based on that range, you have to choose the Vref, very, very important.

Suppose it is giving you 2.2 volts at the maximum and let us say it is giving you 100 millivolts or point one volt at the minimum, you can choose a reference which is at 1.1, it is not going to work because it is going to saturate. Therefore, choice of Vref is a very important design parameter that is number 1. Number 2 is it is always good to choose this Vref as high as possible. On the other side, you can say that it is good to choose a high Vref as well, maybe 3.3 or even 5 volts.

Because if you choose a higher Vref, the SNR signal to noise ratio of the ADC also improves. It is ability to cut off noise is also good at higher references. So, Vref is an important parameter as we go along, I will give you some material to read from on Arduino and so on which will allow you to understand this issue of noise in ADC, Vref in ADC as critical parameters for several of your sensing applications. Now let us complete this discussion on sensor interfacing to IoT device an embedded device.

Let us draw our attention to a important sensor which is basically measuring nitrogen dioxide and that sensor essentially is a semiconductor sensor. It is also measuring carbon monoxide both, it is giving you both.

(Video Starts: 41:54)

And once that sensor is this sensortech is MiCS 4514 may be useful to expand a little bit. As you can see this is a mass sensor with two fully independent sensing elements in one package. Now, you can go into the details of this picture, but before we actually delve into it. What is important is? Suppose that you are interested in measuring the nitrogen dioxide concentration in the air, you have to get to use this chart.

This chart essentially is giving you the nitrogen dioxide. The top one is telling you the cross sensitivity of such a sensor, it is cross sensitive to methane, propane, ammonia and so on and so forth so, many other related things. Of course, it is also measuring carbon monoxide, it is measuring CO. But when you are measuring CO there are other gases also it is cross sensitive. So, one of the problems with any of this air quality measurements by this particular IoT applications and IoT sensors is the cross sensitivity of these sensors.

You have to really be very careful about it. That would after all their electrical engineers will get a voltage. Whether that voltage corresponds to a particular gas that is present in the ambient or whether it is corresponding to a mixture of gases, we do not know because it is only a voltage. Therefore, you have to find a way to subtract the others and then gas of interest you should pull out and so on.

So, they have to do lot of adjustments to actually pull out that particular thing which is standard. I mean, everybody does this, so this chart is telling you that. If you measure CO, you are not sure whether you have measuring H2S, ethanol, hydrogen, ammonia, and so on and so forth. Your environment to a large extent will tell you whether that environment actually is, you know, also rich in other gases. So, you must know your environment.

For example, if you are in an industrial setting its difficult to say what are the gases that are eliminating. But if you are at home, you can be sure that you will not be playing with ammonia for instance right or any one of them. So, you are actually could be measuring just the carbon monoxide present somewhere. Quite like that the that is when you do it basically all these gas sensors particularly the semiconductor gas sensors do this oxidation and reduction process.

So, this is the reduction process which gives you the carbon monoxide and the oxidation process gives you the NO2 gas. But again, NO2 gas here is cross sensitive to NO as well as to hydrogen. Having discussed this cross sensitivity? Suppose you are interested in finding out the PPM level of nitrogen dioxide. How do you do that? This is the only chart available with you. On the x axis, it is a log scale, on the x axis, you see the concentration in PPM or on the y axis you see a ratio of r s divided by r naught.

Now this sensor is supposed to give you r s as well as r naught. Now r naught is something that you have to somehow make one measurement under the clean environment you have to do it. Assuming that there are no gases around you, make a measurement of that r naught value. R naught is nothing but a resistance value you have to get hold of a resistance value which this system is actually giving.

Then r s essentially is the measured value of NO2 that you are interested in making a measurement on that. So, that is an environment where there is the gas of interest. r naught is the environment your placed maybe it is a testing chamber, maybe it is a calibration place, somewhere where you are calibrating the sensor and you are obtaining r naught. Then you take the ratio of r s by r naught you will get a number.

You can see that this number is along the y axis corresponding to that if you read off the slope and you do a simple expression and you read of the look at the slope then you will actually essentially be able to read off the concentration value. So, from this picture itself will be able to read up the concentration value.

(Video Ends: 46:26)

Now let us shift to the way by which this code is implemented, and I will show you in software how exactly this code works.

(Video Starts: 46:31)

On the screen you can see that you are actually obtaining the resistance value which essentially is marked there. So, that mark actually is reading the resistance value r s. And you can see that expression essentially is trying to read that the circuit given by is SGX sensor will actually tell you the references of you will actually that you have to put this expression in order to make a measurement of the r s value.

Repeat the same process to get the r naught value as well into a clean environment where you know that there are no gases there of and that gives you an r naught value. So, together you come back with the r s by r naught into this other data sheet and then make a measurement. So, you can see that the PPM value that you are actually measuring is based on simply that the other graph that I mentioned to you, the graph which I showed you.

You just read off the PPM from the x axis when you know, the r s by r naught which you have actually found out in a painstaking way of done this r naught calculation and you have done the r s by r naught. So, after that you print the PPM value. The gas sensor is ready now to show you all about the PPM values which are being read as the nitrogen dioxide values as it is being correct.

So, you can see now that ADC value of something tells you output voltage is 1.27 sensor resistances is some value and the concentration of nitrogen dioxide is so much PPM. So, you can see continuously it is actually what we are showing you is live in this classroom. You see that ADC value is changing to some extent and corresponding output voltage as given by the sensor is shown there.

The sensor resistance is calculated as we discussed r s is actually being shown there, r naught is already known to you because it is a one-time activity that you do an r naught. So, r s is being displayed then use r s by r naught and put that simple expression to calculate the PPM value. That is about what we have here in this demonstration. Thank you very much.

(Video Ends: 49:03)